Automated Parameter Studies

- **Script-based tool automation for ModelDesk**
- **Automated parameterization and experiment management**
- **Efficiency and convenience for parameter studies**

Long-term tests and parameter studies can now benefit from script-based tool automation in ModelDesk, the parameterization software for the Automotive Simulatin Models (ASM). This offers users maximum flexibility to define custom simulation scenarios, utilizing scripting languages like Python and MATLAB M.

Tool Automation for ModelDesk

ModelDesk 1.1 supports remote control based on script languages such as Python and MATLAB M, and also AutomationDesk, the test automation software from dSPACE. The new feature is available under the name of tool automation and uses ModelDesk's COM interface (Microsoft COM, Common Object Model). It means that all functions for experiment management and vehicle or environment parameterization previously available via the GUI are now also available via a programmable interface. Thus, nearly everything that can be done manually by clicking buttons or entering values can also be executed from a script.

Use Cases for Tool Automation

Tool automation brings efficiency and convenience to parameter studies. As in real test drives, maneuvers can be performed repeatedly, varying certain conditions each time. This allows standard tests to run on a virtual basis before vehicle prototypes are available, and new control strategies to be tested without expensive physical infrastructures:

- Fishhook maneuver/rollover detection
  A short script that covers the following steps will be sufficient: set the test velocity, perform the fishhook maneuver, check if the tire lift exceeds the end condition, if not, increase the test velocity and restart. Once the tire lift condition is reached, the script can end the loop and save the test results.

- ESP corner braking
  Tool automation can also be used to check the vehicle stability controller (ESP) connected to a dSPACE Simulator under different conditions. The corner braking maneuver can be performed on a road repeatedly, with varying road frictions, test velocities, or additional loads on the vehicle, for instance. It is even possible to run the maneuver on corners with different radii by switching between predefined roads.

Script

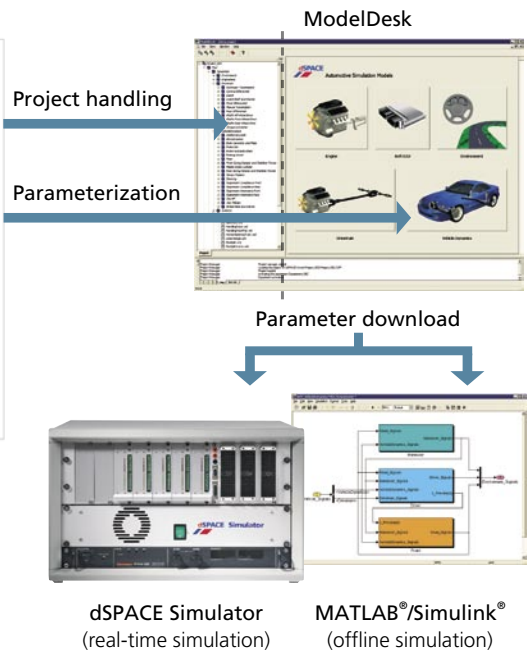
```
TestDescription = [
   ('TestName' :   'Low vehicle mass',
    'Maneuver' :   'ESPBrake',
    'Parameter': { 'Path': 'VD.BGaM.C_m_V',
                   'Value': 1800.0 }),
   ('TestName' :   'High vehicle mass',
    'Maneuver' :   'ESPBrake',
    'Parameter': { 'Path': 'VD.BGaM.C_m_V',
                   'Value': 2200.0 })]
try:
   Initialize(TestEnvironment)

   for Test in TestDescription:
      DownloadManeuver(Test['Maneuver'])
      SetParameter(Test['Parameter'])
      RunManeuver(TestEnvironment)

      GetResults(TestEnvironment)
      EvaluateResults()

finally:
      CleanUp(TestEnvironment)
```

▲ *Simplified extract from a Python script that shows how to define several tests and perform them in a loop.*



Automated Results

The use cases illustrate how ModelDesk's tool automation helps to gather valuable data in early stages of the development process. The feature is seamlessly integrated into the dSPACE tool chain and can be used in offline simulation for developing new control algorithms and also for testing ECUs on a HIL simulator in real time.